



## COMPLIANCE COMPONENT

DEFINITION	
<i>Name</i>	Interoperability Best Practices
<i>Description</i>	“Interoperability” encompasses the tools, techniques, services, and methods that enable the various elements of distributed business applications and systems to interoperate. These elements can share function, content, and communications across heterogeneous computing environments. In particular, interoperability offers a set of architecture services such as platform and service location transparency, transaction management, basic messaging between two points, and guaranteed message delivery.
<i>Rationale</i>	<p>A common, interoperable architecture fosters collaboration among agencies and departments within agencies with similar technologies as they attempt to ensure that standards and technologies have a broad range of applicability across the enterprise. This reduces incidence of ‘orphan’ technologies used only at a single agency.</p> <p>A common set of application integration standards (or interoperable standards) accelerates development of applications by reducing the complexity of the development environment. Such an environment encourages development of re-usable solutions among agencies. It also allows agencies to share training and support resources more easily.</p> <p>Interoperability is necessary to implement resource sharing and Service Oriented Architecture (SOA).</p>
<i>Benefits</i>	<ul style="list-style-type: none"> <li>• Promotes interoperability between systems and organizations.</li> <li>• Creates foundation for future computing.</li> <li>• Creates infrastructure for Service Oriented Architecture (SOA).</li> <li>• Facilitates heterogeneous enterprise architecture.</li> <li>• Couples future services, data and operations.</li> <li>• Reduces Total Cost of Ownership (TCO) and operations.</li> <li>• Increases ease of use for customers.</li> <li>• Moves to shared solutions for business processes and functions.</li> </ul>
ASSOCIATED ARCHITECTURE LEVELS	
<i>Specify the Domain Name</i>	Interoperability
<i>Specify the Discipline Name</i>	
<i>Specify the Technology Area Name</i>	
<i>Specify the Product Component Name</i>	
COMPLIANCE COMPONENT TYPE	
<i>Document the Compliance Component Type</i>	Guideline
<i>Component Sub-type</i>	
COMPLIANCE DETAIL	
<i>State the Guideline, Standard or Legislation</i>	Business systems developed by Missouri agencies should consider the following interoperability best practices:

## **Establish and Communicate a Clear Vision**

A clear vision allows people to work towards a common goal. A clear vision limits the scope of the application. Vision is subject to change. It is important to have a vision document that is maintained as an organization's vision changes. This vision document can be used for many different communication purposes. It is a document that is meant to be distributed and used as an aid in any project. Another best practice is to have a vision document on a per project basis. In establishing this vision document, the project vision should be compared to the organizational vision to assess if the project vision contributes to the overall organizational vision.

## **Define Goals and Objectives**

Goals and objects for applications systems should be maintained at the organizational level as well as a project level. Goals and objects should meet a vision. Goals and objectives must be measurable. When gathering application requirements, goals and objectives can be used to determine if a requirement is within the scope of a project. A requirement should meet an objective, the objective should meet a goal, and the goal should meet a vision.

## **Measure**

When writing a vision, goal, objective or application requirement it is imperative that it is measurable. Measuring and verification insures that a vision, goal, objective or requirement is achieved. Measurements include dates, resources, quality of service, monetary and other financial indicators. Visions, goals, objectives and requirements should be written in a form that includes measurements, and it should be clear how to perform the measurement. When it is not clear, the method of measurement should also be part of the statement.

## **Maintain an External View**

The key to gathering good applications requirements is to maintain an external perspective on an organization. Every activity that is performed in an organization should be done ultimately to serve an external entity. All too often there are activities in organizations that have no benefit to external entities. These activities are usually easy to uncover in gathering application requirements by simply asking the question, "Why?" If the answer is, "Because we have always done it that way", it is quite possible that the purpose of the activity has been long lost, and business processes have subsequently changed leaving the activity without an external reason to exist. This is an orphaned activity with no external purpose and should be eliminated. An external view is also an excellent way to maintain an interoperability and customer-centric view. When visions, goals, objectives and requirements are written from interoperability, customer-centric/external views organizations realize their real need for existence.

## **Develop Iteratively**

The most effective way to manage an application /interoperability project is to manage on an iterative basis. Change should be anticipated, encouraged and managed. Iteratively developing application /interoperability starts with the assumption that interoperability requirement will change and, that iteratively, interoperability application can be brought up-to-date. Iteration also provides a complete feedback loop that incorporates the measurement best practice. When an application /interoperability project is implemented in either an interoperability process or automated in software it can be measured to see that the

	<p>interoperability/business requirement is met. There will be times that the measurement will not be met, and this becomes feedback to the next iteration that allows more analyzed again, with more perfect information and then re-implement.</p> <p><b>Model</b></p> <p>Modeling with a graphical modeling language provides a visual way to analyze, comprehend, organize, optimize, and communicate interoperability requirements. The process of modeling is based on abstraction. All models are abstract. In other words, they lack detail. By graphically modeling interoperability requirements it is possible to focus on the most important details of the interoperability requirements. Additionally, modeling allows several competing arrangements of interoperability requirements to be analyzed, thus allowing better decisions to be made.</p> <p><b>Manage Interoperability and Technology Requirements</b></p> <p>Interoperability and technology requirements change and require management. Change management applies to interoperability requirements as much as it applies to changes to source code in a software development project. Priority of interoperability requirements should be based on the priorities of the organization. Some interoperability requirements may exist because of legal or regulatory requirements while other interoperability requirements may exist for purely financial reasons. Managing interoperability requirements requires assessing the impact of change. When interoperability requirements change, interoperability processes and software that automates these processes should change to meet the changing requirements. There can be significant cost associated to change interoperability processes and software. When change does occur, the changes need to be fully analyzed, designed into solutions and deployed in implementation. Traceability of interoperability requirements is critical to assessing that change. A change in interoperability requirement can cause change in two directions. They can change goals, objectives and vision of an organization, or cause change to software requirements which in turns causes change to the design and implementation of software. Traceability allows changes to be assessed.</p> <p><b>Quality Assurance</b></p> <p>Quality assurance should be an integral part of interoperability requirements gathering and implementation. Undiscovered defects in interoperability requirements are among the most expensive defects possible in systems development. Detection of and repair of an interoperability defect would include gathering software requirements based on a defective interoperability requirement, followed by the analysis of the software requirement, design and implementation of the software requirement and the testing of the implementation.</p>		
<i>Document Source Reference #</i>			
<b>Compliance Sources</b>			
<i>Name</i>		<i>Website</i>	
<i>Contact Information</i>			
<i>Name</i>		<i>Website</i>	
<i>Contact Information</i>			
<b>KEYWORDS</b>			
<i>List Keywords</i>	Interoperability, Service Oriented Architecture, SOA		

COMPONENT CLASSIFICATION			
<i>Provide the Classification</i>	<input type="checkbox"/> <i>Emerging</i>	<input checked="" type="checkbox"/> <i>Current</i>	<input type="checkbox"/> <i>Twilight</i> <input type="checkbox"/> <i>Sunset</i>
<i>Sunset Date</i>			
COMPONENT SUB-CLASSIFICATION			
<b>Sub-Classification</b>	<b>Date</b>	<b>Additional Sub-Classification Information</b>	
<input type="checkbox"/> <i>Technology Watch</i>			
<input type="checkbox"/> <i>Variance</i>			
<input type="checkbox"/> <i>Conditional Use</i>			
Rationale for Component Classification			
<i>Document the Rationale for Component Classification</i>			
Migration Strategy			
<i>Document the Migration Strategy</i>			
Impact Position Statement			
<i>Document the Position Statement on Impact</i>			
CURRENT STATUS			
<i>Provide the Current Status</i>	<input type="checkbox"/> <i>In Development</i>	<input type="checkbox"/> <i>Under Review</i>	<input checked="" type="checkbox"/> <i>Approved</i> <input type="checkbox"/> <i>Rejected</i>
AUDIT TRAIL			
<i>Creation Date</i>	06/23/05	<i>Date Approved / Rejected</i>	10/11/05
<i>Reason for Rejection</i>			
<i>Last Date Reviewed</i>		<i>Last Date Updated</i>	
<i>Reason for Update</i>			

<http://www.bell-labs.com/user/krauscher/nric/>